

APPLICATION FOR UNITED STATES LETTERS PATENT

INVENTORS: Woong Hee CHAE

TITLE: METHOD FOR MANAGING SOCKET IN MOBILE
COMMUNICATION SYSTEM

ATTORNEYS: FLESHNER & KIM, LLP
& P.O. Box 221200
ADDRESS: Chantilly, VA 20153-1200

DOCKET NO.: HI-0056

METHOD FOR MANAGING SOCKET IN MOBILE COMMUNICATION SYSTEM

BACKGROUND OF THE INVENTION

1. Field of the Invention

[1] The present invention relates to a mobile communication system, and more particularly, to a method for managing a socket allocated to a call control processor in a mobile communication system.

2. Description of the Related Art

[2] A socket is used for transmitting data using a specific protocol between a server and a client. The communication between processors via the socket is based on a client-server model. A client processor can communicate with the server processor through the socket connection. For this, the client processor generates an undesignated socket and requests a connection with a socket designated by the server. Therefore, if the connection is successfully completed, each file descriptor is supplied to the server and the client, and both file descriptors can be used for reading and writing.

[3] A general server processor first initializes itself, and then waits for a service request from a client. At this time, a specific port number is allocated by the client for receiving a request from the client. This port is monitored by the server processor. When

the client transmits data to an address and a port of the server, the server shifts out of the waiting status so as to execute the requested service. If the requested service is an interlink type service, a linkage is established and maintained until the linkage is expressly cut off. Since bi-directional transmission is supported through the socket communication, when one socket is opened, both data transmission and reception can be simultaneously executed via the socket.

[4] The above-described socket can be implemented by a mobile communication system using a TCP/IP to periodically manage a status of each process.

[5] Figure 1 is a diagram of a related art mobile communication system that performs call control processing. Referring to Figure 1, the mobile communication system includes a Base Station Manager (BSM) 130, a Radio Network Controller (RNC) 100, and a plurality of Base Stations (BSs) 120a ~ 120n. The BSM 130 is used to manage radio resources in the RNC 100. The RNC 100 receives information from the plurality of BSs 120a ~ 120n. The RNC 100 provides the received information to a higher layer to a core network (not illustrated), and transmits information received from the higher layer to the plurality of the BSs 120a ~ 120n. The plurality of the BSs 120a ~ 120n provide the information received from the RNC 100 to a user, and transmit information inputted from the user to the RNC

100. The information may include voice data, character data, control commands, and the like.

[6] The RNC 100 preferably includes a Call Control Processor (CCP) 110, a Service Control Processor (SCP) 112, a Switch Module Processor (SMP) 114, and a Vocoder Multi-channel Processor (VMP) 116. The CCP 110 is a main processor that controls a call processing of the RNC. The SCP 112 is a processor for managing a signal point, and executes connection management relating to a Number 7 signaling network. Moreover, the SMP 114 is a processor to control selectors in a mobile communication system, and may perform a function of selecting information received from the plurality of the BSs 120a~120n. The VMP 116 is a processor for controlling a vocoder, which performs a function of decoding encoded voice data into PCM data in a mobile communication system.

[7] The CCP 110, which is connected through a Board Support Package (BSP) 121 as a main processor in the plurality of BSs 120a ~120n and the socket, periodically controls the status of the BSP 121. Moreover, the CCP 110 is connected to SCP 112, SMP 114, and VMP 116 in the inner part of the RNC 100, and can control the status of each of the processors 112, 114, 116.

[8] As described above, the CCP 110 requires numerous socket resources because it is connected to each of the processors 112, 114, 116, 121 (hereinafter, the plurality of target

processors) and the socket, and has to control a status of each of the plurality of target processors.

[9] Figure 2 is a flow chart illustrating a related art method for monitoring a status of each processor using the socket. Referring to Figure 2, the CCP 110 allocates a socket to each of the plurality of target processors 112, 114, 116, and 121, so as to periodically control the status of each of the plurality of target processors (Step 201). At this time, each of the plurality of target processors, as well as the socket allocated to the CCP, allocates its own socket for communicating with the CCP 110.

[10] When the socket is allocated to each of the target processors, the CCP 110 may periodically receive a status message from the plurality of target processors. The CCP 110 reports the received status information to the BSM 130. Furthermore, the CCP 110 transmits its own status to each of the plurality of target processors using a message (Step 211). In this case, the CCP 110 transmits the status message in a prescribed form when the plurality of the target processors request a status message via the socket.

[11] Accordingly, each processor's status is periodically communicated between the CCP 110 and the plurality of target processor based on the connection through the socket.

[12] When communicating through the socket, the communication is executed by an application layer via the transport layer, and not via a network layer. Further the

communication is performed in accordance with an Internet Protocol (IP). The IP allows for transmission of data packets between other networks according to the IP address and thus provides path control. IP is typically used with a Transmission Control Protocol (TCP) or a User Datagram Protocol (UDP), which are protocols included in the transmission layer. Because the CCP 110 utilizes the TCP/IP, it is necessary to allocate socket resources to each of the plurality of the target processor.

[13] The socket can be classified into either a stream socket that utilizes TCP or a datagram socket that utilizes the UDP, according to the transmission protocol it utilizes. The UDP's main function is to designate ports of a receiving body and a sending body by a UDP header. The port is a number for discriminating the receiving body and the sending body through a service or a higher protocol in the inner portion of a host that has an IP address. The TCP, on the other hand, provides the port designating function, and further includes a transmission controlling function for controlling a communication sequence, answering back, re-transmission, data stream, urgent data, congestion and the like.

[14] Referring again to Figure 2, when a status message is received from one of the plurality of target processors, the CCP 110, may generate a status change command based on the status messages. The CCP 110 then transmits the status change command to the corresponding processor (Step 221). The corresponding processor is then able to change its

status according to the status change command.

[15] The related art socket management has numerous problems. For example, as described above, the status of other processors between the CCP and the plurality of target processors can be monitored by periodically exchanging the status thereof. However, in the related mobile communication system using the IP, numerous socket resources are required because the CCP must allocate a socket to each of the plurality of target processors to be communicated with.

[16] Moreover, in the related CCP, if a socket is allocated to a processor that is not mounted or is not properly functioning, a socket is allocated to a target processor which should not be allocated. Accordingly, socket resources are wasted. Specifically, although the disconnected target processor is not monitored when disconnected communication occurs in the related CCP, the socket resource is not withdrawn.

[17] Moreover, the status messages are continuously and repeatedly transmitted to the plurality of the target processors in the related art CCP, causing the loads in the CCP to be increased.

[18] Additionally, when a socket is allocated to a processor which does not need to be communicated with or is continuously allocated for the transmission of a status message because the CCP fails to withdraw the allocated socket resources, call control processing may

not be properly executed if the socket resource is not corrected or withdrawn.

[19] The above references are incorporated by reference herein where appropriate for appropriate teachings of additional or alternative details, features and/or technical background.

SUMMARY OF THE INVENTION

[20] An object of the invention is to solve at least the above problems and/or disadvantages and to provide at least the advantages described hereinafter.

[21] Another object of the present invention is to provide a method for managing a socket to monitor a status of a target processor based on whether a message transmitted from the target processor is received or not.

[22] Another object of the present invention is to provide a method for managing a socket to release the socket allocation if an error occurs at a target processor.

[23] It is another object of the present invention to provide a method for managing a socket to release the socket in accordance with a reception of a message transmitted from the target processor.

[24] It is another object of the present invention to provide a method for managing a socket CCP according to a status of a target processor in a radio network controller RNC.

[25] To achieve at least the above objects in whole or in parts, there is provided a method for a socket management in a mobile communication system, including allocating a socket corresponding to a call control processor in a plurality of target processors, communicating with the plurality of target processor using the allocated socket and releasing the socket allocation to the corresponding target processor depending on the reception of the message.

[26] To achieve at least the above objects, in whole or in parts, there is further provided a method for a socket management in a mobile communication system, including communicating with a plurality of target processors utilizing a socket allocated corresponding to the plurality of target processors, transmitting a status confirmation message to the target processor which has received the message depending on the receipt of the message and executing a socket management based on the response to the status confirmation message.

[27] To achieve at least the above objects, in whole or in parts, there is further provided a method of managing a socket in a mobile communication system, including allocating a socket between a call control processor (CCP) and each of a plurality of target processors, sending a request status message from the CCP to each of the target processors using the corresponding socket, sending a status confirmation message from the CCP to

non-responding target processors, and one of reallocating the socket to the non-responding processors and releasing the socket to the non responding processor.

BRIEF DESCRIPTION OF THE DRAWINGS

[28] The invention will be described in detail with reference to the following drawings in which like reference numerals refer to like elements wherein:

[29] Figure 1 is a diagram illustrating a related art mobile communication system for call control processing;

[30] Figure 2 is a flow chart illustrating a related art method for monitoring a status of each processor using a socket; and

[31] Figure 3 is a flow chart illustrating a method for managing a socket allocated corresponding to each processor in a mobile communication system where the status of each processor is monitored using the socket according to a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[32] Figure 3 is a flow chart showing a method for managing an allocated socket corresponding to each processor in a mobile communication system where the status of

each processor is monitored using the socket, according to a preferred embodiment of the present invention. It is noted that the preferred embodiment is described with reference to the block diagram of the mobile communication system in Figure 1, in which the same reference numbers represent the same elements. Moreover, the method according to the present invention is preferably realized through the use of a control block, a socket communication device, and the like, within the CCP. Although, the control block and the socket communication device are hardware-like terms, the terms are used to emphasize those functions work at the software level. Further description of the device or system is omitted.

[33] Referring to Figure 3, CCP 110 preferably allocates a socket to each of the plurality of target processors 112, 114, 116, and 121, so as to periodically monitor the status of each of the plurality of the target processors (Step 301). At this time, each of the plurality of the target processors, as well as the socket allocated to the CCP 110, preferably allocates its own socket to communicate with the CCP 110.

[34] When the socket is allocated to each of the plurality of target processors, the CCP 110 periodically receives status messages from the plurality of the target processors. The CCP 110 reports the status of target processors to the BSM 130. Furthermore, the CCP 110 transmits its status to the plurality of target processors through a status message (Step

304). The CCP 110 preferably transmits the status message in a prescribed form when there is a request, via the socket, for the status message from the plurality of target processors.

[35] Accordingly, each processor's status is periodically communicated between the CCP 110 and the plurality of target processor based on the connection through the socket.

[36] Next, it is determined whether the status message has been received from the plurality of target processors (Step 307). That is, the CCP 110 requests the status message from each of the plurality of target processors according to a prescribed order. The CCP 110 next checks to determine whether a status message has been received from each of the plurality of target processors within a prescribed period of time. If the status message is received from each of the plurality of target processors within the prescribed period of time, then the CCP 110 determines that the target processor that has sent the status message is in normal state, and transmits a status change command generated from the status message to the target processor. The CCP 110 thus directs the status of the processor to be changed according to the status change command (Step 310).

[37] If, however, the status message has not been received from a given one of the target processors within the prescribed time period, the CCP 110 transmits a status confirmation message to the target processor using UDP (Step 316). The state

confirmation message is preferably transmitted by utilizing UDP through the port of the sender and the recipient designated by the UDP header.

[38] The CCP 110 next checks to determine whether a response to the status confirmation message transmitted to the target processor has been received or not.

[39] If a response is received to the status confirmation message from the target processor, the CCP 110 determines that there is an error on the socket allocated to the target processor and re-allocates the socket to the target processor after withdrawing the existing socket (Step 322).

[40] If, however, the CCP 110 receives no response to the status confirmation message from the target processor, the CCP determines that a down state has occurred in the target processor, and releases the socket that has been allocated to that target processor (Step 325).

[41] The above description relates to a method for managing a socket to release particular target processors, but whenever it is confirmed that the status message has not been received from any one of the plurality of the target processor, the CCP 110 may release the allocated socket in the target processor by transmitting the status message based on the response of the status confirmation message. In addition, it is noted that the socket allocated to the target processor could always be re-used on another target processor.

[42] The present invention as broadly described herein has many advantages. For

example, by releasing the socket allocation based on the receipt of the response on the status confirmation message and re-allocating the released socket to other target processor, the usage efficiency of the socket resource and the system performance can be enhanced.

[43] Additionally, because a status message is no longer transmitted to the socket-released target processor, the load in the CCP is reduced, thus enhancing efficiency.

[44] The foregoing embodiments and advantages are merely exemplary and are not to be construed as limiting the present invention. The present teaching can be readily applied to other types of apparatuses. The description of the present invention is intended to be illustrative, and not to limit the scope of the claims. Many alternatives, modifications, and variations will be apparent to those skilled in the art. In the claims, means-plus-function clauses are intended to cover the structures described herein as performing the recited function and not only structural equivalents but also equivalent structures.